

31000000 11254

PERANCANGAN KOMPENSATOR JARINGAN SYARAF TIRUAN DI OPERATIONAL SPACE UNTUK MENGATASI KETIDAKPASTIAN PADA MANIPULATOR ROBOT

TUGAS AKHIR

Disusun oleh :

TRICAHYA WIRATNO

NRP : 2292 100 020

RSE

629.892

Wir

P-1

1998

PERPUSTAKAAN ITS	
Tgl. Terima	13-12-99
Terima Dari	N 8880
No. Agenda Pzp.	



JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA

1998

PERANCANGAN KOMPENSATOR JARINGAN SYARAF TIRUAN DI OPERATIONAL SPACE UNTUK MENGATASI KETIDAKPASTIAN PADA MANIPULATOR ROBOT

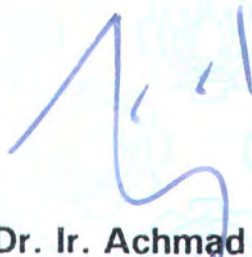
TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik Elektro
Pada**

**Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya**

Mengetahui/Menyetujui

Dosen Pembimbing



Dr. Ir. Achmad Jazidie, M.Eng
NIP. 131 647 788

**SURABAYA
MARET, 1998**

PERANCANGAN SISTEM KENDALIAN ROBOT MELAKUKAKAN PERGERAKAN SUKSES DALAM RUANG 3D UNTUK MENUNTUTKAN TUGAS PADA MANIPULATOR ROBOT

HENRI ANHAR

Departemen Teknik Elektro dan Teknologi Informasi
Universitas Indonesia, Jalan Sekeloa Selatan 1
Depok, 16159, Indonesia

Bidang Studi Sistem dan Proses Industri
Jurusan Teknik Elektro
Fakultas Teknik Industri
Universitas Teknologi Sepuluh Nopember
Surabaya

Mengototasi Mengetahui

Design Engineering

On or Before 10th May 2010
Ref: 131-4-7-100

Abstrak

Perancangan sistem kendali robot untuk melakukan pergerakan
suksesan dalam ruang 3D untuk menuntutkan tugas pada
manipulator robot. Perancangan sistem kendali robot dilakukan
dengan menggunakan metode perancangan sistem kendali robot.

ABSTRAK

Salah satu metoda yang digunakan untuk pengendalian gerakan manipulator robot adalah metoda *computed torque*. Metoda ini akan berhasil dengan baik jika parameter-parameter dari manipulator robot diketahui dengan tepat. Namun pada kenyataannya, untuk dapat mengetahui parameter-parameter tersebut secara tepat sangatlah sulit, sehingga menimbulkan ketidakpastian (*uncertainty*) pada pengendalian manipulator robot, yang mengakibatkan kesalahan pada posisi *end-effector* manipulator lengan robot. Banyak metoda yang diajukan untuk mengatasi ketidakpastian tersebut. Salah satunya adalah dengan menggunakan bantuan jaringan syaraf tiruan. Namun, penggunaan jaringan syaraf tiruan pada metoda-metoda tersebut menyebabkan masalah lain timbul, seperti performansi jaringan yang digunakan dan kesulitan kita untuk dapat membayangkan pergerakan dari manipulator robot karena metoda-metoda tersebut biasanya bekerja pada *joint space*. Pada tugas akhir ini, dicoba dikembangkan suatu metoda untuk mengatasi ketidakpastian tersebut. Metoda yang diusulkan ini dilakukan dengan menambahkan kompensator jaringan syaraf tiruan pada pengendalian manipulator robot dengan menggunakan metoda *computed torque* yang akan bekerja pada *operational space*. Selanjutnya, untuk menguji efektifitas dan validitas dari metoda yang dikembangkan ini dilakukan simulasi dengan menggunakan komputer.



Kata Pengantar

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Puji syukur kami panjatkan kepada Allah Yang Maha Pengasih dan Maha Penyayang, yang telah melimpahkan rahmat dan petunjuk-Nya, sehingga penulis dapat menyelesaikan tugas akhir sebagai salah satu syarat memperoleh gelar sarjana Jurusan teknik Elektro.

Dalam menyusun tugas akhir ini, penulis mendapat bimbingan serta bantuan atas saran dari berbagai pihak. Dan atas terselesaikannya tugas akhir ini penulis ingin menyampaikan rasa terima kasih yang sedalam-dalamnya kepada :

- Bapak Dr. Ir. Achmad Jazidie M.Eng. , selaku Dosen Pembimbing dalam penyusunan tugas akhir ini yang telah banyak sekali memberikan arahan, nasehat, bimbingan, dan semangat kepada penulis.
- Bapak Dr. Ir. Moch. Rameli, selaku Koordinator Bidang Studi Teknik Sistem Pengaturan yang telah banyak memberikan fasilitas kepada penulis.
- Bapak dan ibu dosen pada Bidang Studi Teknik Sistem Pengaturan pada khususnya dan Jurusan Teknik Elektro pada umumnya.
- Almarhum babe, mama yang berjuang demi keluarga sepeninggal babe, Dian, Mas Indro, Mas Adhi beserta Putri dan Kiki, Om Heni, Irma yang

jauh, serta saudara-saudara penulis lainnya yang memberikan bantuan dan dorongan kepada penulis.

- Rekan-rekan di Bidang Studi Teknik Sistem Pengaturan, Zul, Yusuf, Kojar, Ancha, Toni, Johan, Yana, Felix, Okim, Dodo, Abed, MAM, Aris, Huda, Pris, Bram, Gundul, Daus, dan para alumni, Adi, Romli, Cucun, Wiku, Wambe, juga rekan-rekan lainnya yang tidak mungkin tersebutkan seluruhnya di sini, atas semangat dan segala dukungannya.
- Mustain, Bambang, Winahyu, Didik 'lemu', Gendul, dan rekan-rekan angkatan E-32 yang lain, yang selalu setia memberikan hiburan, dukungan, dan gangguan yang semua sangat berarti bagi penulis.
- Dan rekan-rekan yang telah memberikan kenangan manis maupun kenangan pahit yang tidak akan terlupakan.

Besar harapan penulis semoga tugas akhir ini bermanfaat bagi para pembaca pada umumnya serta mahasiswa Teknik Elektro pada khususnya. Penulis menyadari segala kekurangan yang ada sehingga tugas akhir ini masih jauh dari kesempurnaan, oleh karena itu kritik dan saran selalu penulis harapkan.

Akhirnya sekali lagi puji syukur ke hadirat Allah SWT.

Wassalamu'alaikum Wr. Wb.

Surabaya, Maret 1998

Penulis

Daftar Isi

DAFTAR ISI

	halaman
Abstrak	ii
Kata Pengantar	iii
Daftar Isi	v
Daftar Gambar	viii
Daftar Tabel	ix
BAB I	
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Pembatasan Masalah	3
1.4 Tujuan	3
1.5 Sistematika Pembahasan	4
BAB II	
DINAMIKA ROBOT	5
2.1 Formulasi Newton-Euler	5
2.2 Formulasi Lagrangian	7
2.3 Persamaan Dinamik Manipulator Robot Pada <i>Joint Space</i>	8
2.3.1 Matriks Massa Manipulator	9
2.3.2 Bentuk Sentrifugal Dan Koreolis	10

2.4	Dinamika Manipulator Robot Pada <i>Operational Space</i>	11
-----	--	----

BAB III

ALGORITMA BACK-PROPAGATION

JARINGAN SYARAF TIRUAN	15
3.1 Model Sel Syaraf (Neuron)	15
3.2 Arsitektur Jaringan Syaraf Tiruan	18
3.3 Algoritma Belajar	18
3.3.1 Algoritma Back-Propagation	20
3.3.2 Tahapan Pada Algoritma Back-Propagation	21
3.3.3 Propagasi Maju	22
3.3.4 Propagasi Balik	23

BAB IV

KOMPENSATOR JARINGAN SYARAF TIRUAN	28
4.1 Metoda Computed Torque	29
4.2 Kompensator Jaringan Syaraf Tiruan	31
4.3 Struktur Jaringan Syaraf Tiruan	34
4.4 Cara Kerja Kompensator Jaringan Syaraf Tiruan	35

BAB V

EKSPERIMEN DENGAN SIMULASI	37
---	----

BAB VI

PENUTUP	49
6.1 Kesimpulan	49



6.2	Saran	50
	DAFTAR PUSTAKA	51
	LAMPIRAN	52

Daftar Gambar

DAFTAR GAMBAR

	halaman
Gambar 3.1 Model Sel Syaraf (Neuron)	16
Gambar 3.2 Output Fungsi Sigmoid	17
Gambar 3.3 Struktur Lapisan Hidden	18
Gambar 4.1 Sistem Pengendalian Robot Menggunakan Metoda Computed Torque	29
Gambar 4.2 Sistem Pengendalian Robot Menggunakan Metoda Computed Torque Ditambah Kompensator JST	32
Gambar 4.3 Cara Mendapatkan Teaching Signal	33
Gambar 4.4 Struktur Jaringan Syaraf Tiruan Yang Digunakan	35
Gambar 5.1 Lengan Robot Planar 2 Link	37
Gambar 5.2 Grafik Posisi X Simulasi I	41
Gambar 5.3 Grafik Posisi Y Simulasi I	42
Gambar 5.4 Grafik Posisi X Simulasi II	44
Gambar 5.5 Grafik Posisi Y Simulasi II	45
Gambar 5.6 Grafik Posisi X Simulasi III	47
Gambar 5.7 Grafik Posisi Y Simulasi III	48

Daftar Tabel

DAFTAR TABEL

	halaman
Tabel 5.1 Parameter Lengan Robot	38
Tabel 5.2 Hasil Simulasi I	40
Tabel 5.3 Hasil Simulasi II	43
Tabel 5.4 Hasil Simulasi III	46
Tabel 5.5 Kesalahan Posisi End-Effector Hasil Simulasi II	49
Tabel 5.6 Kesalahan Posisi End-Effector Hasil Simulasi III	50

Bab I

BAB I

PENDAHULUAN

1.1. Latar Belakang

Perkembangan teknologi robot sangatlah pesat. Manipulator robot telah menjadi suatu kebutuhan yang penting dalam industri di negara-negara maju, karena mempunyai kecepatan dan ketepatan serta ketahanan yang tidak dimiliki oleh seorang manusia dalam melakukan tugasnya. Namun demikian, untuk dapat mengendalikan manipulator robot sesuai dengan tugas yang harus dilakukannya secara akurat tidaklah mudah. Untuk itu diperlukan suatu *controller* yang didasarkan pada suatu metoda yang dapat memodelkan dinamika dari manipulator robot secara tepat. Ketidakakuratan yang terjadi pada pengendalian manipulator robot disebabkan karena metoda yang digunakan pada *controller* tidak dapat memodelkan semua parameter yang mempengaruhi dinamika dari manipulator robot tersebut. Atau dengan kata lain, *controller* tersebut hanya dapat memodelkan parameter-parameter tertentu yang kita ketahui yang dapat kita ukur, seperti panjang dan massa dari lengan robot. Sedangkan parameter-parameter yang tidak dapat kita ketahui atau yang tidak dapat kita ukur dengan tepat, seperti inersia, akan termodelkan dalam *controller* dengan tidak tepat. Artinya, model yang kita gunakan untuk mendesain *controller* tersebut tidak sesuai dengan dinamika manipulator robot yang sesungguhnya.

Salah satu metoda untuk mengendalikan pergerakan dari manipulator robot adalah metoda *computed torque*. Metoda ini akan bekerja dengan baik jika semua parameter-parameter manipulator robot diketahui dengan tepat. Namun pada kenyataannya, parameter-parameter tersebut sangat sulit untuk dapat diketahui secara tepat, yang dapat menimbulkan ketidakpastian (*uncertainty*). Untuk mengatasi permasalahan tersebut, pada tugas akhir ini dicoba diusulkan suatu metoda dengan menggunakan kompensator jaringan syaraf tiruan yang berfungsi untuk mengkompensasi kesalahan-kesalahan yang timbul karena hal tersebut di atas.

1.2. Permasalahan

Telah banyak dikembangkan metoda dengan menggunakan jaringan syaraf tiruan untuk mengatasi ketidakpastian tersebut. Antara lain, *Atkeson* [1], *Jordan* [2], dan *Psaltis* [3], yang memperkenalkan suatu metoda untuk mendapatkan model invers dinamik dari manipulator robot dalam jaringan syaraf tiruan. Juga metoda yang diajukan oleh *Li* dan *Slotine* [4], yang menggunakan metoda *self learning* untuk mendapatkan model invers dinamik dari manipulator robot. Namun masih terdapat kelemahan dan masalah yang timbul pada metoda-metoda tersebut. Antara lain performansi dari jaringan syaraf tiruan yang digunakan tidak begitu baik, karena jaringan syaraf tiruan tersebut digunakan untuk mempelajari keseluruhan dari sistem sehingga bekerja terlalu berat, yang akan mengakibatkan lamanya proses belajar dari jaringan syaraf tiruan tersebut.

Masalah lain yang timbul adalah, karena sebagian besar metoda-metoda tersebut bekerja pada *joint space*, sehingga timbul kesulitan untuk memahami / membayangkan pergerakan *end-effector* dari manipulator robot tersebut.

1.3. Pembatasan Masalah

Metoda kompensator jaringan syaraf tiruan ini dicoba untuk diterapkan pada program simulasi dinamik manipulator lengan robot yang telah ada. Di mana metoda ini akan disimulasikan untuk mengatasi masalah ketidakpastian yang disebabkan karena ketidaktepatan pengetahuan kita terhadap parameter dari manipulator lengan robot tersebut.

Untuk membandingkan hasil dari penerapan metoda ini, dilakukan juga simulasi dengan menggunakan metoda *computed torque*, dengan parameter yang tidak tepat sesuai dengan parameter sesungguhnya, maupun dengan parameter yang tepat.

1.4. Tujuan

Metoda yang dikembangkan ini bertujuan untuk dapat mengatasi adanya ketidakpastian (*uncertainty*) yang disebabkan karena ketidaktepatan pengetahuan kita terhadap parameter-parameter manipulator robot yang dapat menyebabkan kesalahan dalam pergerakan *end-effector* dari manipulator robot tersebut.

Selain itu, metoda yang dikembangkan ini diharapkan dapat meningkatkan performansi dari jaringan syaraf tiruan yang digunakan, dan lebih memudahkan

pemahaman terhadap pergerakan *end-effector* dari manipulator robot, karena metoda ini akan bekerja tidak lagi pada *joint space* namun pada *operational space*.

1.5. Sistematika Pembahasan

Pada tugas akhir ini, penulis menyusun buku ini dalam beberapa bagian pembahasan. Urutan pembahasan yang dilakukan adalah sebagai berikut.

Latar belakang pengambilan judul, permasalahan yang dihadapi, batasan permasalahan, tujuan yang akan dicapai, dan sistematika pembahasan dibahas pada bab 1.

Pada bab 2, dibahas teori yang berkaitan dengan dinamika dari manipulator robot. Di mana pada bab ini dijelaskan faktor-faktor yang berpengaruh pada persamaan gerak manipulator robot.

Pada bab 3, dibahas mengenai teori jaringan syaraf tiruan, mengenai model, arsitektur, dan algoritma belajar *back-propagation*.

Sedangkan pada bab 4, dibahas tentang metoda yang diusulkan, yaitu kompensator jaringan syaraf tiruan. Pada bab ini juga dibahas mengenai sistem pengendalian dari manipulator robot.

Bab 5 membahas tentang hasil dari simulasi yang telah dilakukan pada kasus manipulator lengan robot planar dengan 2 buah link.

Akhirnya, dari pembahasan dan simulasi yang telah dilakukan dapat diambil kesimpulan yang diberikan pada bab 6. Pada bab ini juga diberikan saran-saran yang mungkin berguna bagi pengembangan lebih lanjut dari tugas akhir ini.

Bab II

BAB II

DINAMIKA ROBOT

Perilaku dinamik manipulator lengan robot dinyatakan dalam kecepatan perubahan konfigurasi lengan dalam hubungannya dengan torsi *joint* yang dihasilkan oleh *actuator*. Hubungan ini dapat dinyatakan dalam sekelompok persamaan diferensial, yang disebut persamaan gerak (*equations of motion*), yang menghasilkan respon dinamik hubungan lengan terhadap input torsi *joint*. Ada beberapa metode yang bisa digunakan untuk mendapatkan persamaan gerak, yaitu formulasi *Newton-Euler*, dan formulasi *Langrangian*.

2.1. Formulasi Newton-Euler

Formulasi Newton-Euler diturunkan dari interpretasi langsung hukum Newton kedua tentang gerak, yang menyatakan sistem dinamik dalam gaya dan momentum. Persamaan tersebut menghubungkan semua gaya dan momen yang bekerja pada suatu *link* lengan, termasuk gaya dan momen kopel diantara *link*. Persamaan yang didapatkan dengan metode Newton-Euler melibatkan gaya-gaya pembatas yang bekerja diantara *link-link* yang berdekatan. Oleh karena itu, operasi matematik tambahan diperlukan untuk mengeliminasi dan mendapatkan hubungan yang eksplisit antara torsi *joint* dan resultan gerak sebagai fungsi pergerakan *joint*.

Gerakan benda tegar dapat dipecah menjadi gerakan translasional dari titik sembarang yang tetap terhadap benda tegar, dan gerakan rotasional dari benda tegar di sekitar titik tersebut. Persamaan dinamik dari benda tegar juga dapat dinyatakan dengan dua persamaan, yaitu yang menggambarkan gerakan translasional dari titik pusat massa (*centroid*), dan yang menggambarkan gerakan rotasional di sekitar *centroid*. Yang pertama adalah persamaan gerak Newton untuk partikel massa, dan yang kedua disebut persamaan gerak Euler.

Gaya inersia dinyatakan dengan $-m_i \dot{v}_{ci}$, di mana m_i adalah massa *link* dan \dot{v}_{ci} adalah turunan waktu dari v_{ci} . Selanjutnya persamaan gerak didapatkan dengan menambahkan gaya pada kesetimbangan gaya statis, sehingga

$$f_{i-1,i} - f_{i,i+1} + m_i g - m_i \dot{v}_{ci} = 0 \quad i = 1, \dots, n \quad (2.1)$$

di mana,

$f_{i-1,i}$ dan $-f_{i,i+1}$ adalah gaya kopel yang diberikan pada *link* i oleh *link* $i-1$ dan $i+1$, dan g adalah percepatan gravitasi.

Gerakan rotasional diperoleh dari persamaan Euler. Dengan cara yang sama pada gerakan translasional, persamaan dinamik diturunkan dengan menambahkan torsi inersial pada kesetimbangan momen statis. Dimulai dengan menyatakan massa dari benda tegar dalam kaitannya dengan gerak rotasi disekitar *centroid*. Massa tersebut dinyatakan dengan *inertia tensor*, yang merupakan matriks simetris 3 x 3.

Torsi inersial yang bekerja pada *link* i dinyatakan sebagai kecepatan perubahan momentum angular dari tiap *link* tiap satuan waktu. ω_i adalah vektor kecepatan *angular* dan I_i adalah *inertia tensor centroidal* dari *link* i , sehingga

momentum didapatkan dari $I_i \omega_i$. Karena *inertia tensor* berubah-ubah oleh perubahan orientasi *link*, maka turunan waktu dari momentum *angular* tidak hanya memasukkan percepatan *angular* $\dot{I}_i \omega_i$, tetapi juga melibatkan faktor yang dihasilkan dari perubahan *inertia tensor*. Faktor tersebut disebut *torsi gyroscopic* dan dinyatakan dengan $\omega_i \times (I_i \omega_i)$. Dengan menambahkan faktor tersebut pada kesetimbangan momen sebelumnya, maka menghasilkan

$$\mathbf{N}_{i-1,i} - \mathbf{N}_{i,i} + \mathbf{r}_{i,i} \times \mathbf{f}_{i,i-1} - \mathbf{r}_{i-1,i} \times \mathbf{f}_{i-1,i} - \dot{\mathbf{I}}_i \omega_i - \omega_i \times (\mathbf{I}_i \omega_i) = 0 \quad i = 1, \dots, n \quad (2.2)$$

Persamaan tersebut menggambarkan perilaku dinamik dari suatu *link* lengan individual. Persamaan lengkap dari keseluruhan manipulator didapatkan dengan menerapkan kedua persamaan di atas untuk semua *link* lengan.

2.2. Formulasi Lagrangian

Dalam formulasi Lagrangian, perilaku dinamik sistem dinyatakan dalam kerja dan energi dengan menggunakan koordinat yang digeneralisasi. Semua gaya yang tidak melakukan kerja dan gaya-gaya pembatas otomatis dapat dieliminasi di dalam metoda ini.

Formulasi Lagrangian menjelaskan persamaan gerak dinamik berdasarkan energi dan usaha menggunakan koordinat general. Jika q_1, q_2, \dots, q_n adalah koordinat general dan T dan U adalah total energi kinetik dan energi potensial pada sistem dinamik, maka kita dapat mendefinisikan Lagrangian L dengan,

$$L(q_i, \dot{q}_i) = T - U \quad (2.3)$$

Persamaan gerak dari sistem dinamik menggunakan Lagrangian,

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q \quad i = 1, \dots, n \quad (2.4)$$

Q_i adalah gaya general pada koordinat general q_i . Gaya general dapat diidentifikasi sebagai *virtual work* yang bekerja dengan gaya *non-conservative* yang bekerja pada sistem.

Dengan menggunakan kedua metoda di atas akan didapatkan persamaan dinamik lengan robot seperti yang ditunjukkan pada persamaan berikut,

$$\sum_{j=1}^n M_{ij} \ddot{q}_j + \sum_{j=1}^n \sum_{k=1}^n h_{ijk} \dot{q}_j \dot{q}_k + G_i = \tau \quad i = 1, \dots, n \quad (2.5)$$

di mana M adalah *inertia tensor* dan h_{ijk} adalah

$$h_{ijk} = \frac{\partial M_{ij}}{\partial q_k} - \frac{1}{2} \frac{\partial M_{jk}}{\partial q_i} \quad (3.2.4)$$

dan

$$G_i = \sum_{j=1}^n m_j g^T J_{Li}^j \quad (2.6)$$

2.3. Persamaan Dinamik Manipulator Robot Pada *Joint Space*

Manipulator lengan robot dimodelkan sebagai suatu kumpulan dari n gerakan benda rigid yang dihubungkan dalam suatu rangkaian seri dengan ujung yang satu terpasang pada bumi dan ujung yang lainnya bebas. Benda rigid (*link*)

tersebut terhubung bersama dengan sebuah *joint*, di mana pada tiap *joint* terdapat torsi aktuator dan gaya gesek.

Persamaan (2.5) dapat ditulis dalam bentuk persamaan yang lebih umum dipakai, yaitu pada persamaan (2.7) berikut ini,

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + G(q) \quad (2.7)$$

di mana,

τ adalah vektor $n \times 1$ dari torsi *joint*,

q adalah vektor $n \times 1$ dari posisi *joint*,

M adalah matriks $n \times n$ dari massa manipulator,

V adalah vektor yang mewakili torsi yang timbul akibat dari gaya sentrifugal dan *coreolis*, dan

G adalah vektor yang mewakili torsi yang timbul dari gravitasi.

Karena adanya ketidakpastian (*uncertainty*), maka persamaan (2.7) akan menjadi,

$$\tau = M(q)\ddot{q} + V(q, \dot{q}) + G(q) + T_d \quad (2.8)$$

Di mana T_d adalah vektor dari sinyal yang tidak diketahui yang timbul karena dinamika manipulator yang tidak termodelkan dan parameter manipulator yang tidak tepat.

Berikut ini akan dijelaskan mengenai faktor-faktor yang berpengaruh pada persamaan dinamik manipulator robot, yaitu faktor yang berasal dari massa manipulator robot, faktor dari gaya sentrifugal dan *coreolis*, serta faktor dari adanya gaya gravitasi.

2.3.1. Matriks Massa Manipulator

Manipulator robot mempunyai energi kinetik yang dapat dituliskan sebagai berikut,

$$E = \frac{1}{2} \dot{q}^T K(q) \dot{q} \quad (2.9)$$

di mana,

$K(q)$ adalah matriks yang menggambarkan distribusi massa dari suatu manipulator sebagai fungsi dari vektor *joint* q , di mana matriks ini harus definit positif sehingga harga dari persamaan di atas selalu positif dan mewakili energi kinetik.

Kemudian kita dapat menuliskan energi potensial $P(q)$ dari suatu manipulator dengan suatu fungsi skalar dari posisi *joint*. Sehingga *Lagrangian* dari sistem adalah sebagai berikut,

$$L = \frac{1}{2} \dot{q}^T K(q) \dot{q} - P(q) \quad (2.10)$$

Dengan menggunakan metode Lagrange yang telah dijelaskan di atas maka akan didapatkan persamaan dinamik sebagai berikut,

$$\frac{d}{dt} \left(K(q) \dot{q} \right) - \frac{\partial}{\partial q} \left(\frac{1}{2} \dot{q}^T K(q) \dot{q} - P(q) \right) = \tau \quad (2.11)$$

atau dapat ditulis,

$$K(q) \ddot{q} + K(q) \ddot{q} - \frac{1}{2} \dot{q}^T \left(\frac{\partial}{\partial q} K(q) \right) \dot{q} + \frac{\partial}{\partial q} P(q) = \tau \quad (2.12)$$

Sehingga jika kita bandingkan persamaan (2.5) dengan persamaan (2.12), maka didapatkan bahwa matriks massa manipulator $M(q)$ sama dengan matriks energi kinetik $K(q)$.

2.3.2. Bentuk Sentrifugal dan *Coreollis*

Kita dapat menuliskan bentuk sentrifugal dan *coreollis* ini dalam beberapa bentuk, melalui beberapa penurunan yang berbeda. Dengan menyamakan bentuk dari persamaan (2.8) dan persamaan (2.12) dalam q , maka diperoleh,

$$V(q, \dot{q}) = \dot{M}(q)\dot{q} - \frac{1}{2}\dot{q}^T \left(\frac{\partial}{\partial q} K(q) \right) \quad (2.13)$$

di mana,

$$\dot{M}(q) = 2V_m(q, \dot{q}) - J \quad (2.14)$$

Sehingga kita dapat menuliskan vektor sentrifugal dan *coreollis* dalam bentuk berikut,

$$V(q, \dot{q}) = V_m(q, \dot{q})\dot{q} \quad (2.15)$$

Selain itu kita juga dapat menuliskan vektor sentrifugal dan *coreollis* dalam bentuk berikut ini,

$$V(q, \dot{q}) = \begin{bmatrix} \dot{q}^T V_1(q) \dot{q} \\ \dot{q}^T V_2(q) \dot{q} \\ \vdots \\ \dot{q}^T V_n(q) \dot{q} \end{bmatrix} \quad (2.16)$$

di mana $V_i(q)$ adalah matriks simetris $n \times n$.

2.4. Persamaan Dinamik Manipulator Robot Pada *Operational Space*

Untuk memudahkan penentuan posisi dan orientasi dari *end-effector*, diperlukan suatu persamaan yang menggambarkan dinamika dari *end-effector* tersebut. Persamaan dinamik tersebut sebagai fungsi dari variabel-variabel pada *end-effector*, atau dengan kata lain persamaan dinamik tersebut bekerja pada *operational space*. Persamaan dinamik tersebut akan menentukan gaya yang diperlukan untuk menggerakkan *end-effector* ke posisi dan orientasi yang diinginkan.

Kita dapat menyamakan usaha-usaha yang terjadi pada *joint* dengan usaha-usaha yang terjadi pada *end-effector* dengan cara sebagai berikut, kita memiliki

$$F \cdot \delta x = \tau \cdot \delta q \quad (2.17)$$

di mana,

F adalah vektor 6×1 dari gaya pada *end-effector*,

δx adalah vektor 6×1 dari pergeseran (*displacement*) dari *end-effector*,

τ adalah vektor 6×1 dari torsi pada *joint*, dan δq adalah vektor 6×1 dari pergeseran *joint*.

Persamaan (2.17) dapat juga ditulis sebagai,

$$F^T \delta x = \tau^T \delta q \quad (2.18)$$

Sedangkan *Jacobian* adalah,

$$\delta x = J \delta q \quad (2.19)$$

dari persamaan (2.19) dan persamaan (2.17) diperoleh,

$$F^T J \delta q = \tau^T \delta q \quad (2.20)$$

$$F^T J = \tau^T \quad (2.21)$$

atau,

$$\tau = J^T F \quad (2.22)$$

Transpose *Jacobian* pada persamaan (2.22) di atas memetakan gaya pada *end-effector* dengan torsi pada *joint* yang bersesuaian.

Hubungan antara variabel *joint space* dan variabel *operational space* adalah sebagai berikut,

$$\dot{x} = J \dot{q} \quad (2.23)$$

$$\ddot{x} = \dot{J} \dot{q} + J \ddot{q} \quad (2.24)$$

Dengan menggunakan persamaan (2.23), persamaan (2.24) di atas, dan persamaan (2.8) maka persamaan dinamik manipulator pada *operational space* akan dapat kita turunkan sebagai berikut,

$$J^T F = M(q) \ddot{q} + V(q, \dot{q}) + G(q) + T_d \quad (2.25)$$

Dengan mengalikan kedua sisi persamaan dengan $JM^{-1}(q)$, maka didapatkan

$$JM^{-1}(q) J^T F = J \ddot{q} + JM^{-1}(q) V(q, \dot{q}) + JM^{-1}(q) G(q) + F_d \quad (2.26)$$

Dengan memasukkan persamaan (2.24), didapatkan

$$JM^{-1}(q) J^T F = \ddot{x} - \dot{J} \dot{q} + JM^{-1}(q) V(q, \dot{q}) + JM^{-1}(q) G(q) + F_d \quad (2.27)$$

sehingga,

$$F = (JM^{-1}(q)J^T)^{-1} \ddot{x} + (JM^{-1}(q)J^T)^{-1} \left\{ JM^{-1}(q)V'(q, \dot{q}) - \dot{J}\dot{q} \right\} + (JM^{-1}(q)J^T)^{-1} JM^{-1}(q)G(q) + F_d \quad (2.28)$$

sehingga didapatkan persamaan dinamik dari manipulator yang bekerja pada *operational space* sebagai berikut,

$$F = M_x(x) \ddot{x} + V_x(x, \dot{x}) + G_x(x) + F_d \quad (2.29)$$

di mana,

$$M_x(x) = (JM^{-1}(q)J^T)^{-1} \quad (2.30)$$

$$V_x(x, \dot{x}) = (JM^{-1}(q)J^T)^{-1} \left\{ JM^{-1}(q)V(q, \dot{q}) - \dot{J}\dot{q} \right\} \quad (2.31)$$

$$G_x(x) = (JM^{-1}(q)J^T)^{-1} JM^{-1}(q)G(q) \quad (2.32)$$

dan F_d adalah vektor yang mewakili ketidakpastian yang timbul karena adanya dinamika yang tidak termodelkan pada manipulator.

Bab III

BAB III

ALGORITMA BACK-PROPAGATION

JARINGAN SYARAF TIRUAN

Jaringan Syaraf tiruan dibentuk berdasarkan kerja jaringan syaraf otak manusia. Otak manusia tersusun dari banyak sel syaraf yang disebut *neuron* dan saling terhubung. Otak manusia juga mempunyai karakteristik mengolah informasi yang masuk secara paralel, mempunyai kemampuan belajar, kemampuan mengelola dirinya dan mengingat informasi yang masuk terdahulu.

Jaringan syaraf tiruan menurut Kohonen, didefinisikan sebagai berikut :

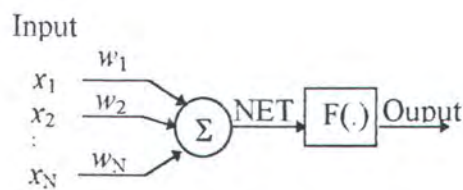
Suatu jaringan terhubung paralel yang biasanya terdiri dari elemen-elemen adaptif dan organisasi hirarkisnya, yang diharapkan dapat berinteraksi dengan benda atau keadaan nyata sama seperti sistem syaraf atau otak biologis melakukannya.

Pada kenyataannya, tidak semua karakteristik dari otak manusia dapat ditiru oleh jaringan syaraf tiruan. Berikut ini akan dibahas mengenai model dari jaringan syaraf tiruan yang digunakan.

3.1. Model Sel Syaraf (*Neuron*)

Gambar berikut menunjukkan model dari *neuron* yang digunakan dalam jaringan syaraf tiruan. Tiga elemen dasar dari model *neuron* tersebut adalah,

penghubung (*synapses*) yang berbobot, penjumlah (*adder*), dan fungsi aktivasi. Input suatu *neuron* (x_1, x_2, \dots, x_N) dapat berupa data dari luar jaringan atau merupakan output dari *neuron* yang lain, yang kemudian dikalikan dengan bobot dari sinapsisnya, w_i .



Gambar 3.1.
Model sel syaraf (*Neuron*).

Anggaplah gambar di atas merupakan unit *neuron* lapis j yang diberi pola input X sebanyak i , dan bias $\theta = 1$, maka secara matematis output Y akan berbentuk

$$NET_j = \sum_{i=0}^N x_{ij} w_{ij} + \theta_{j1} \quad (3.1)$$

$$Y = f(NE T_j) \quad (3.2)$$

dengan w adalah elemen-elemen pembobot input dan $f(.)$ merupakan fungsi aktivasi keluaran jaringan.

Input X_i yang masuk ke *neuron* dapat berupa informasi dari luar sistem atau dari output lapisan *neuron* sebelumnya (lapisan *neuron* yang hirarkinya lebih rendah), jika jaringan syaraf tiruan tersebut memiliki banyak lapisan *neuron* (multi layer). Untuk jaringan syaraf tiruan multilapis, maka hanya lapisan *neuron* dengan hirarki paling rendah (lapisan input) yang menerima sinyal informasi eksternal.

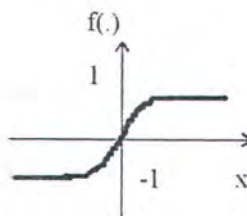
Pembobot w_{ji} pada model di atas, menyatakan kekuatan hubungan antara *neuron-neuron* lapisan ke- i ke lapisan ke- j . Jika *neuron* pada lapisan ke- i berjumlah n , dan pada lapisan j berjumlah m , maka pembobot w_{ji} akan berbentuk matrik $m \times n$. Penulisan *subskrip* pada W selalu dimulai dari lapisan hirarki tertinggi, sehingga untuk model tersebut di atas ditulis sebagai w_{ji} .

Semua data input yang telah diberi pembobot dengan bobot sambungannya akan dijumlahkan terlebih dahulu sebelum diolah menjadi sinyal output. Hasil penjumlahan tersebut biasanya dinotasikan sebagai *NET*.

Fungsi yang mengolah data input menjadi data output disebut fungsi output atau fungsi aktivasi *neuron* $f(.)$. Fungsi aktivasi yang digunakan dalam tugas akhir ini adalah fungsi *sigmoid* berikut,

$$f(x) = \frac{2}{(1 + e^{-x})} - 1 \quad (3.3)$$

Bentuk output dari fungsi *sigmoid* di atas terlihat pada gambar sebagai berikut ini,

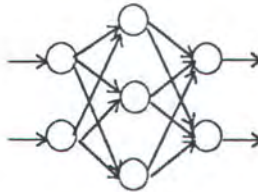


Gambar 3.2.

Output fungsi *threshold sigmoid*.

3.2. Arsitektur Jaringan Syaraf Tiruan

Arsitektur jaringan syaraf tiruan yang digunakan pada tugas akhir ini adalah jaringan syaraf tiruan multilapis (*multilayer*), yang memiliki lapisan *hidden* yang terletak antara lapisan input dengan lapisan output. Berdasarkan arah aliran informasinya, jaringan syaraf tiruan yang digunakan adalah jaringan umpan maju (*feedforward networks*). Arsitektur jaringan syaraf tiruan tersebut seperti terlihat pada gambar berikut ini,



Gambar 3.3.
Struktur lapisan *hidden*.

Jaringan umpan maju mempunyai struktur hirarki di mana antara *neuron-neuron* dalam satu lapisan dengan lapisan sebelumnya, tersambung dalam arah aliran informasi dari tingkat yang lebih rendah maju ke lapisan berikutnya atau yang hirarkinya lebih tinggi. Jadi antara output dengan input tidak terjadi aliran informasi yang berbalik, atau umpan balik.

3.3. Algoritma Belajar

Belajar (*learning*) bagi jaringan syaraf tiruan merupakan proses mengatur harga-harga dari parameter bobotnya untuk mendapatkan hasil yang terbaik, dengan pelatihan (*training*), sesuai dengan unjuk kerja sistem yang dikehendaki.

Tujuan dari belajar ini adalah agar set pola input yang diberikan menghasilkan pola output yang diinginkan atau paling tidak mendekati.

Pelatihan ini dibentuk dengan menerapkan secara berurutan pola vektor input dan mengatur pembobot jaringan agar mendekati pola vektor output, dengan mengikuti suatu algoritma belajar tertentu. Selama proses belajar, pembobot secara perlahan konvergen menuju harga tertentu, sehingga pola input menghasilkan pola output yang diinginkan. Algoritma belajar yang digunakan dalam tugas akhir ini adalah algoritma belajar secara *supervised*, yang memerlukan pasangan pola input dan output.

Metoda belajar yang ada sekarang ini dikembangkan dari konsep yang disusulkan oleh Hebb, di mana pembobot akan ditambah jika *neuron* sumber dan *neuron* tujuan diaktifkan. Hal ini analog dengan memperkuat lintasan-lintasan dalam jaringan. Secara matematis dapat dituliskan sebagai berikut.

$$w_{ij}(n+1) = w_{ij}(n) + \eta y_i y_j \quad (3.4)$$

di mana,

$w_{ij}(n)$ adalah harga pembobot dari *neuron i* ke *neuron j* sebelum diatur.

$w_{ij}(n+1)$ adalah harga pembobot dari *neuron i* ke *neuron j* setelah diatur.

η adalah koefisien *learning rate*, y_i adalah keluaran dari *neuron i* dan merupakan input dari *neuron* ke *j*, dan y_j adalah output dari *neuron j*.

3.3.1. Algoritma Back-Propagation

Algoritma belajar *back-propagation*, pertama kali diformulasikan oleh Werbos dan dipopulerkan oleh Rumelhart dan McClelland untuk dipakai pada jaringan syaraf tiruan, didisain untuk beroperasi pada jaringan *feedforward* multilapis dengan metoda *supervised*.

Algoritma ini banyak dipakai pada aplikasi pengendalian karena prosedur belajarnya didasarkan pada hubungan yang sederhana, sebagai berikut :

Jika output memberikan hasil yang salah maka pembobot dikoreksi supaya error dapat diperkecil dan respon jaringan selanjutnya diharapkan akan lebih mendekati harga yang benar.

Prosedur ini analog dengan proses pengaturan pada suatu sistem pengaturan. Algoritma *back-propagation* menangani pelatihan pembobot pada lapisan *hidden*-nya. Tetapi kekurangan utama pada algoritma ini adalah tidak adanya jaminan untuk memperoleh hasil yang konvergen, tetapi karena perhitungan matematisnya yang sederhana algoritma ini tetap banyak dipakai.

Fungsi aktivasi output yang biasa dipergunakan adalah jenis fungsi *sigmoid* atau fungsi *hiperbolik*, karena metoda *back-propagation* memerlukan jenis fungsi yang dapat diturunkan (*differentiable*) dan bersifat *monotone nondecreasing*. Penambahan harga aktivasi hanya dapat menambah output atau tetap, tanpa pernah berkurang.

Penguatan pada fungsi ini merupakan kemiringan kurva pada suatu tingkat eksitasi tertentu, dan nilainya berubah dari harga yang kecil pada eksitasi negatif

besar menjadi harga yang besar pada eksitasi 0 dan kemudian kembali mengecil pada harga eksitasi yang semakin besar dan positif.

3.3.2. Tahapan Pada Algoritma Back-Propagation

Algoritma *back-propagation* ini terdiri dari dua tahap, yaitu tahapan propagasi maju (*forward propagation*) dan tahapan propagasi balik (*backward propagation*). Tahapan propagasi maju dimulai dengan memberikan suatu pola masukan pada lapisan input. Pada lapisan input ini, pola masukan tadi hanya dilewatkan untuk kemudian dikalikan dengan pembobot yang menghubungkannya dengan lapisan di atasnya (lapisan *hidden*). Lapisan input ini merupakan lapisan pasif karena tidak mengolah pola masukan yang diterimanya. Dalam setiap lapisan yang berurutan (kecuali lapisan input), setiap elemen pengolah (*neuron*) menjumlahkan setiap masukannya dan melewatkannya pada fungsi aktivasi untuk mendapatkan outputnya. Output ini kemudian didistribusikan maju ke lapisan berikutnya, untuk kemudian mengalami proses yang sama sampai pada lapisan output. Lapisan output lalu menghasilkan output jaringan secara keseluruhan.

Tahapan propagasi balik dimulai dengan membandingkan respon jaringan keseluruhan dengan pola output yang diinginkan. Jika terdapat perbedaan (*error*), maka *error* tersebut akan digunakan untuk memperbaiki harga pembobot jaringan, dengan mendistribusikannya kembali ke lapisan sebelumnya.

3.3.3. Propagasi Maju

Tahapan ini dimulai dengan memberikan suatu pola masukan pada jaringan.

Misalnya untuk satu vektor masukan berikut,

$$x_p = (x_{p1}, x_{p2}, \dots, x_{pN})^T \quad (3.5)$$

Tiap unit input akan mendistribusikan pola masukan tersebut ke *neuron* pada lapisan berikutnya (*hidden*). *NET* input untuk unit *hidden* ke *j* adalah sebagai berikut,

$$net_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h \quad (3.6)$$

di mana w_{ji}^h adalah pembobot sambungan dari unit input ke *i*, dan θ_j^h adalah bias.

Subskrip *h* mengacu pada kuantitas pada lapisan tersembunyi (*hidden*). Anggaplah aktivasi dari node ini sama dengan *Net* input, maka output dari node ini adalah sebagai berikut,

$$i_{pj} = f_j^h(net_{pj}^h) \quad (3.7)$$

sehingga persamaan output menjadi,

$$net_{pk}^o = \sum_{j=1}^L w_{kj}^o i_{pj} + \theta_k^o \quad (3.8)$$

$$o_{pk} = f_k^o(net_{pk}^o) \quad (3.9)$$

dengan superskrip *o* mengacu pada kuantitas pada lapisan output.

3.3.4. Propagasi balik

Tahapan propagasi balik dimulai dengan membandingkan output jaringan keseluruhan dengan output yang diinginkan. *Error* yang terjadi digunakan untuk memperbarui tiap pembobot jaringan.

Error ini diminimisasikan oleh persamaan berikut,

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (3.10)$$

dengan $\delta_{pk} = (y_{pk} - o_{pk})$ adalah *error* pada lapisan output. Pergerakan harga E_p ini sering juga dipandang sebagai suatu permukaan pada ruang pembobot (*weight space*).

Arah perubahan bobot adalah gradien negatif dari E_p , yang dihitung dengan memperhatikan harga bobot w_{kj} . Perubahan pembobot terjadi sesuai dengan arah gradien negatif tersebut, yang merupakan arah dari *steepest descent* pada titik z pada suatu permukaan ruang pembobot dengan tinggi permukaan sama dengan *error* yang terukur. Kemudian harga pembobot dapat diatur secara iteratif supaya E_p dapat berkurang atau mencapai titik minimum Z_{min} pada ruang pembobot tersebut.

Berikut ini penurunan mekanisme *back-propagation* dalam memperbaiki harga tiap pembobotnya. Persamaan di atas dapat ditulis sebagai berikut,

$$E_p = \frac{1}{2} \sum_k (y_{pk} - o_{pk})^2 \quad (3.11)$$

maka gradien negatif dari E_p terhadap harga pembobot w_{kj} adalah,

$$\frac{\partial E_p}{\partial w_{kj}^o} = -(y_{pk} - o_{pk}) \frac{\partial f_k^o}{\partial (net_{pk}^o)} \frac{\partial (net_{pk}^o)}{\partial w_{kj}^o} \quad (3.12)$$

di mana,

$$\frac{\partial (net_{pk}^o)}{\partial w_{kj}^o} = \left(\frac{\partial}{\partial w_{kj}^o} \sum_{j=1}^L w_{kj}^o i_{pj} + \theta_k^o \right) = i_{pj} \quad (3.13)$$

sehingga diperoleh gradien negatif sebagai berikut,

$$\frac{\partial E_p}{\partial w_{kj}^o} = -(y_{pk} - o_{pk}) f_k^{o'}(net_{pk}^o) i_{pj} \quad (3.14)$$

dengan $f_k^{o'}(net_{pk}^o) = \frac{\partial f_k^o}{\partial (net_{pk}^o)}$

Pembobot jaringan pada lapisan output diperbaharui mengikuti persamaan berikut,

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \Delta p w_{kj}^o(t) \quad (3.15)$$

di mana $\Delta p w_{kj}^o = \eta (y_{pk} - o_{pk}) f_k^{o'}(net_{pk}^o)$

dan η adalah konstanta belajar, yang biasanya berharga lebih kecil dari 1 dan positif.

Dengan menganggap fungsi aktivasi output yang digunakan dalam menurunkan algoritma ini adalah fungsi *sigmoid*,

$$f_k^o(net_{jk}^o) = \left(1 + \exp^{-net_{jk}^o} \right)^{-1} \quad (3.16)$$

yang mempunyai bentuk turunan sebagai berikut,

$$f_k^{o'} = f_k^o(1 - f_k^o) = o_{pk}(1 - o_{pk}) \quad (3.17)$$

maka persamaan untuk mengubah harga pembobot di atas menjadi sebagai berikut,

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta(y_{pk} - o_{pk})o_{pk}(1 - o_{pk})i_{pj} \quad (3.18)$$

Sehingga jika didefinisikan,

$$\begin{aligned} \delta_{pk}^o &= (y_{pk} - o_{pk})f_k^{o'}(net_{pk}^o) \\ &= \delta_{pk}f_k^{o'}(net_{pk}^o) \end{aligned} \quad (3.19)$$

maka persamaan untuk memperbaharui harga pembobot dapat ditulis sebagai berikut,

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta\delta_{pk}^o i_{pj} \quad (3.20)$$

Pembobot pada lapisan tersembunyi (*hidden*) diperbaiki mirip dengan pada lapisan output, namun dengan harga output target yang tidak diketahui. Secara intuitif, total *error* E_P dapat diketahui berhubungan dengan harga output pada lapisan ini, sehingga didapatkan

$$E_P = \frac{1}{2} \sum_k (y_{pk} - o_{pk})^2 \quad (3.21)$$

$$E_P = \frac{1}{2} \sum_k (y_{pk} - f_k^o(net_{pk}^o))^2 \quad (3.22)$$

$$E_P = \frac{1}{2} \sum_k \left(y_{pk} - f_k^o \left(\sum_j w_{kj}^o i_{pj} + \theta_k^o \right) \right)^2 \quad (3.23)$$

Telah diketahui bahwa i_{pj} tergantung pada pembobot lapisan *hidden*, sehingga gradien E_P dengan mengacu pada pembobot lapisan *hidden* dapat dihitung sebagai berikut,

$$\frac{\partial E_P}{\partial w_{ji}^h} = \frac{1}{2} \sum_k \frac{\partial}{\partial w_{ji}^h} (y_{pk} - o_{pk})^2 \quad (3.24)$$

$$\frac{\partial E_p}{\partial w_{ji}^h} = - \sum_k (y_{pk} - o_{pk}) \frac{\partial o_{pk}}{\partial (net_{pk}^o)} \frac{\partial (net_{pk}^o)}{\partial i_{pj}} \frac{\partial i_{pj}}{\partial (net_{pj}^h)} \frac{\partial (net_{pj}^h)}{\partial w_{ji}^h} \quad (3.25)$$

karena,

$$\frac{\partial o_{pk}}{\partial (net_{pk}^o)} = f^{o'}(net_{pk}^o)$$

$$\frac{\partial (net_{pk}^o)}{\partial i_{pj}} = w_{kj}^o$$

$$\frac{\partial i_{pj}}{\partial (net_{pj}^h)} = f^{h'}(net_{pj}^h)$$

$$\frac{\partial (net_{pj}^h)}{\partial w_{ji}^h} = x_{pi}$$

maka persamaan di atas dapat ditulis sebagai berikut,

$$\frac{\partial E_p}{\partial w_{ji}^h} = - \sum_k (y_{pk} - o_{pk}) f^{o'}(net_{pk}^o) w_{kj}^o f^{h'}(net_{pj}^h) x_{pi} \quad (3.26)$$

Pembobot pada lapisan *hidden* diperbaharui sebanding dengan gradien negatif dari E_p , atau harga negatif dari persamaan tersebut,

$$\Delta w_{ji}^h = \eta f^{h'}(net_{pj}^h) x_{pi} \sum_k (y_{pk} - o_{pk}) f^{o'}(net_{pk}^o) w_{kj}^o \quad (3.27)$$

dengan η tetap sebagai *learning rate*. Karena harga δ_{pk}^o telah diketahui, maka persamaan tersebut di atas dapat ditulis sebagai berikut,

$$\Delta w_{ji}^h = \eta f^{h'}(net_{pj}^h) x_{pi} \sum_k \delta_{pk}^o w_{kj}^o \quad (3.28)$$

Terlihat bahwa perbaikan setiap pembobot pada lapisan *hidden* tergantung pada harga *error* δ_{pk}^o dari lapisan output.

Adanya mekanisme inilah yang membuat algoritma ini juga dikenal dengan istilah *Error back-propagation*. Mekanisme ini juga memperlihatkan kemampuannya untuk melatih lapisan *hidden* dari jaringan dan juga menghindari keterbatasan jaringan dengan satu lapisan.

Dengan mendefinisikan *error* pada lapisan *hidden* sebagai berikut,

$$\delta_{pj}^h = f^{h'}(net_{pj}^h) \sum_k \delta_{pk}^o w_{kj}^o \quad (3.29)$$

maka persamaan untuk mengubah harga pembobot pada lapisan *hidden* dapat ditulis sebagai berikut,

$$w_{ji}^h(t+1) = w_{ji}^h(t) + \eta \delta_{pj}^h x_i \quad (3.30)$$

Bab IV

BAB IV

KOMPENSATOR JARINGAN SYARAF TIRUAN

Kompensator menurut definisinya adalah suatu komponen tambahan yang ditambahkan dalam suatu sistem untuk merubah perilaku dari sistem yang tidak memenuhi performa yang diinginkan.

Pada pengendalian manipulator lengan robot dengan menggunakan algoritma kontrol yang berdasarkan pada metoda *computed torque*, output yang dihasilkan biasanya tidak memenuhi kriteria seperti yang diinginkan. Hal tersebut disebabkan karena metoda *computed torque* hanya akan berhasil dengan baik jika parameter-parameter dari manipulator lengan robot tersebut diketahui secara tepat. Pada kenyataannya, terdapat parameter-parameter yang tidak dapat diketahui besarnya dengan tepat, yang akhirnya akan menimbulkan kesalahan pada output, yang berupa posisi dari *end-effector* lengan robot.

Kompensator jaringan syaraf tiruan akan ditambahkan pada pengendalian manipulator lengan robot tersebut untuk dapat mengkompensasi kesalahan yang terjadi. Algoritma belajar yang digunakan pada kompensator jaringan syaraf tiruan ini menggunakan algoritma belajar *back-propagation*, seperti yang telah dijelaskan pada bab sebelumnya.

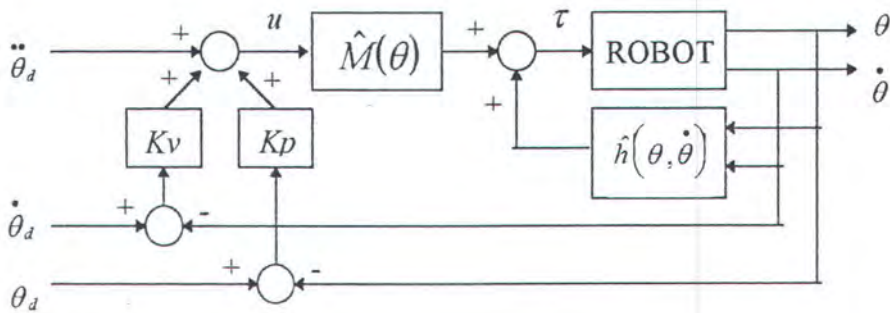
4.1. Metoda Computed Torque

Seperti yang telah dijelaskan pada bab sebelumnya tentang dinamika dari manipulator lengan robot, persamaan gerak dari suatu manipulator lengan robot yang terdiri dari n link adalah sebagai berikut,

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) \quad (4.1)$$

di mana τ adalah $n \times 1$ vektor dari torsi yang dihasilkan oleh *actuator* pada *joint*. M dan h adalah $n \times n$ matriks inersia dan $n \times 1$ vektor torsi yang dihasilkan oleh gaya sentrifugal dan koreolis.

Pengendalian manipulator robot dengan menggunakan metoda *computed torque* dapat dilihat pada gambar berikut,



Gambar 4.1. Sistem pengendalian robot menggunakan metoda *computed torque*.

Pada gambar di atas, besarnya torsi yang digunakan untuk menggerakkan manipulator robot diperoleh dengan persamaan berikut,

$$\tau = \hat{M}(\theta)u + \hat{h}(\theta, \dot{\theta}) \quad (4.2)$$

di mana $\hat{M}(\theta)$ dan $\hat{h}(\theta, \dot{\theta})$ adalah nilai estimasi dari parameter $M(\theta)$ dan $h(\theta, \dot{\theta})$ yang sebenarnya. Sedangkan sinyal kontrol u diperoleh dari,

$$u = \ddot{\theta}_d + K_v \dot{e} + K_p e \quad (4.3)$$

di mana,

$$e = \theta_d - \theta, \text{ dan}$$

$$\dot{e} = \dot{\theta}_d - \dot{\theta}$$

Sedangkan K_v dan K_p adalah gain yang berupa matriks diagonal $n \times n$.

Jika estimasi terhadap parameter manipulator robot adalah benar, di mana $\hat{M} = M$ dan $\hat{h} = h$, maka dari persamaan (4.1) sampai (4.3) diperoleh,

$$\hat{M}(\ddot{\theta}_d + K_v \dot{e} + K_p e) + \hat{h} = M\ddot{\theta} + h \quad (4.4)$$

sehingga didapatkan persamaan *error* berikut,

$$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad (4.5)$$

Dari persamaan *error* di atas, dengan memilih parameter K_v dan K_p yang sesuai, maka akan didapatkan *error steady state* yang sama dengan nol, yang berarti sistem akan stabil.

Pada kenyataannya, parameter yang tepat dari manipulator sulit untuk ditentukan. Sehingga timbul suatu faktor yang tidak pasti yang disebut ketidakpastian (*uncertainty*). Dengan adanya ketidakpastian tersebut, maka persamaan dinamik dari manipulator menjadi,

$$\tau = M(\theta)\ddot{\theta} + h(\theta, \dot{\theta}) + T \quad (4.6)$$

di mana T adalah faktor ketidakpastian.

Dari persamaan (4.1), (4.3), dan (4.6) didapatkan,

$$\hat{M}(\ddot{\theta}_d + K_v \dot{e} + K_p e) + \hat{h} = M\ddot{\theta} + h + T \quad (4.7)$$

Jika \hat{M}^{-1} ada, maka persamaan *error* akan menjadi,

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{M}^{-1}(\Delta M \ddot{\theta} + \Delta h + T) \quad (4.8)$$

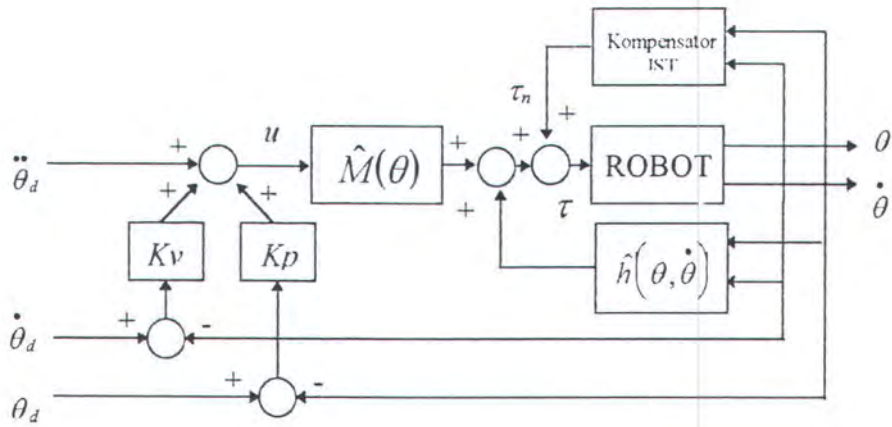
di mana, $\Delta M = M - \hat{M}$ dan $\Delta h = h - \hat{h}$.

Dari persamaan *error* (4.8) di atas, terlihat bahwa *error steady state* tidak akan nol yang berarti sistem tidak akan stabil. Hal tersebut disebabkan karena adanya ketidakpastian seperti yang telah dijelaskan di atas.

4.2 Kompensator Jaringan Syaraf Tiruan

Kompensator jaringan syaraf tiruan akan digunakan bersama-sama dengan model dari manipulator untuk mengkompensasi ketidakpastian yang muncul seperti yang telah dijelaskan pada bab sebelumnya. Sistem pengendalian manipulator robot dengan menggunakan kompensator jaringan syaraf tiruan terlihat seperti pada gambar berikut 4.2.

Seperti telah dijelaskan di atas, kompensator jaringan syaraf tiruan yang digunakan ini menggunakan algoritma *back-propagation* dalam proses belajarnya, di mana algoritma ini memerlukan *teaching signal* sebagai output targetnya.



Gambar 4.2. Sistem pengendalian manipulator robot menggunakan metoda *computed torque* yang menggunakan kompensator jaringan syaraf tiruan.

Teaching signal tersebut dapat diperoleh dengan cara membandingkan besarnya torsi yang dihasilkan oleh sistem kontrol pada gambar 4.1 dengan besarnya torsi yang dihasilkan oleh model manipulator.

Misalkan output dari kompensator jaringan syaraf tiruan adalah τ_N , maka persamaan *error* dari sistem kontrol pada gambar 4.2 dapat diturunkan sebagai berikut,

$$\hat{M}(\ddot{\theta} + K_v \dot{e} + K_p e) + \hat{h} + \tau_N = M\ddot{\theta} + h + T \quad (4.9)$$

Sehingga persamaan errornya akan menjadi,

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{M}^{-1}(\Delta M \ddot{\theta} + \Delta h + T - \tau_N) \quad (4.10)$$

Persamaan *error* (4.10) di atas akan menghasilkan *error steady state* nol jika sisi kanan persamaan tersebut sama dengan nol.

Jika kita menggunakan *teaching signal* jaringan syaraf tiruan sebagai berikut,

$$\tau_t = \Delta M \ddot{\theta} + \Delta h + T \quad (4.11)$$

maka sisi kanan persamaan (4.10) akan menjadi nol.

Besarnya torsi output dari model adalah sebagai berikut,

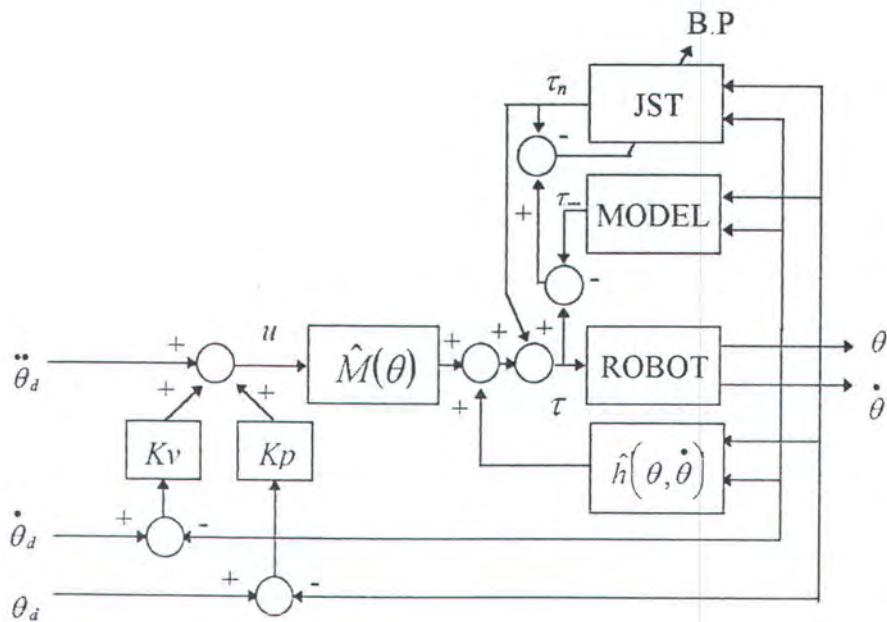
$$\tau_m = \hat{M} \ddot{\theta} + \hat{h} \quad (4.12)$$

Dengan mengurangkan persamaan (4.6) dan (4.12), maka didapatkan

$$\tau - \tau_m = \Delta M \ddot{\theta} + \Delta h + T \quad (4.13)$$

Sehingga dari persamaan (4.13) dan alasan yang telah diuraikan di atas, didapatkan *teaching signal* dari kompensator jaringan syaraf tiruan sebagai berikut,

$$\tau_t = \tau - \tau_m \quad (4.14)$$



Gambar 4.3. Cara mendapatkan *teaching signal* bagi kompensator jaringan syaraf tiruan.

4.3. Struktur Jaringan Syaraf Tiruan

Struktur jaringan syaraf tiruan yang digunakan adalah sebagai berikut,

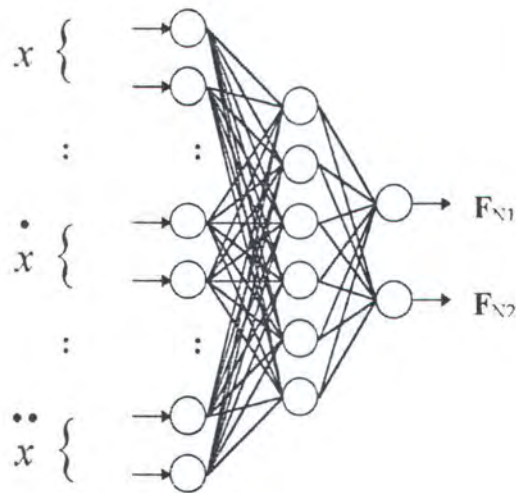
- Mempunyai tiga lapis, lapisan input, *hidden*, dan output.
- Jumlah *neuron* lapisan input adalah 18 buah, dengan input x, \dot{x} , dan \ddot{x} . Di mana x adalah vektor 6 x 1 dari posisi *end-effector*, \dot{x} adalah vektor 6 x 1 dari kecepatan *end-effector*, dan \ddot{x} adalah vektor 6 x 1 dari percepatan *end-effector* manipulator lengan robot. Vektor \ddot{x} didapatkan dari persamaan,

$$\ddot{x}(n) = \frac{\dot{x}(n) - \dot{x}(n-1)}{T_s} \quad (4.15)$$

di mana T_s adalah *sampling period* yang dipilih sekecil mungkin agar harga $\ddot{x}(n)$ mendekati harga yang sebenarnya.

- Jumlah *neuron* lapisan *hidden* adalah 6 buah.
- Jumlah *neuron* lapisan output sesuai dengan jumlah *link* dari lengan robot. Untuk contoh kasus pada buku ini digunakan manipulator lengan robot dengan 2 *link*, sehingga jumlah *neuron* lapisan output adalah 2 buah. Dengan output berupa gaya yang diberikan pada tiap *link*, $F_x(n)$.

Struktur jaringan syaraf tiruan yang telah dijelaskan di atas dapat digambarkan pada gambar 4.4 berikut ini,



Gambar 4.4. Struktur jaringan syaraf tiruan yang digunakan

4.4. Cara Kerja Kompensator Jaringan syaraf Tiruan

Pada simulasi yang dilakukan, lengan robot digerakkan dari titik awal ke titik tujuan selama waktu yang telah ditentukan. Selama pergerakan itu, kompensator jaringan syaraf tiruan akan bekerja untuk mengkompensasi ketidakpastian yang terjadi, seperti yang telah dijelaskan pada bab-bab sebelumnya. Cara kerja dari kompensator jaringan syaraf tiruan tersebut adalah sebagai berikut,

1. Dapatkan informasi posisi, kecepatan, dan percepatan, (x, \dot{x} , dan \ddot{x}) dari *end-effector* dari lengan robot yang bergerak. Lalu kirim informasi tersebut ke *neuron* pada lapisan input jaringan syaraf tiruan.
2. Dapatkan target output (*teaching signal*) jaringan syaraf tiruan dari persamaan (4.14) yang telah diubah ke bentuk *operational space* menjadi,

$$F_t = F_{total} - F_m \quad (4.16)$$

di mana, F_{total} adalah hasil penjumlahan dari gaya yang dihasilkan untuk menggerakkan robot dengan metoda *computed torque*, dengan output dari jaringan syaraf tiruan. Sedangkan F_m adalah gaya yang diperoleh dari model manipulator lengan robot.

3. Dilakukan perhitungan pada tahap propagasi maju (*forward propagation*), sampai didapatkan output dari jaringan syaraf tiruan.
4. Bandingkan output jaringan syaraf tiruan tersebut dengan output targetnya. Jika *error*-nya masih lebih besar dari toleransi yang diijinkan, maka dilakukan perhitungan pada tahap propagasi balik (*backward propagation*).
5. Ulangi langkah ke-3, sampai didapatkan *error* yang diperoleh dibawah toleransi yang diijinkan.
6. Tambahkan output jaringan syaraf tiruan, dengan gaya yang digunakan untuk menggerakkan lengan robot.

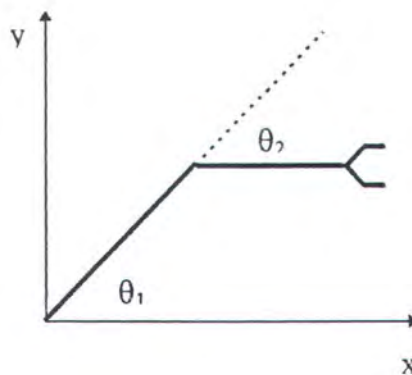
Langkah-langkah di atas diulangi sampai dengan lengan robot tersebut selesai bergerak.

Bab V

BAB V

EKSPERIMEN DENGAN SIMULASI

Simulasi yang dilakukan adalah menggerakkan *end-effector* manipulator lengan robot dari satu titik ke titik tujuan tertentu melalui suatu trajektori / lintasan selama waktu yang telah ditentukan. Lengan robot yang digunakan adalah *planar* (2 dimensi) yang mempunyai 2 buah *link*, seperti pada gambar berikut ini.



Gambar 5.1. Lengan robot planar 2 *link*.

Konfigurasi awal dari lengan robot tersebut adalah, $\theta_1 = 45^\circ$ dan $\theta_2 = -45^\circ$. Dengan posisi awal *end-effector* adalah,

$$X^d(0) = [0.3368(\text{m}) \quad 0.1768(\text{m})]^T$$

Lengan robot digerakkan selama selang waktu 3 detik menuju posisi akhir yang diharapkan, yaitu

$$X^d(3) = [0.35507(\text{m}) \quad 0.205(\text{m})]^T$$

Parameter-parameter lengan robot yang sebenarnya adalah sebagai berikut,

Tabel 5.1. Parameter lengan robot.

<i>Link</i>	Panjang (m)	Massa (kg)	Inersia
1	0.25	9.5	4.3
2	0.16	5.0	1.6

Untuk membuktikan apa yang telah diuraikan pada bab-bab sebelumnya, dilakukan tiga macam simulasi, yaitu :

1. Simulasi menggerakkan lengan robot tersebut dengan metoda *computed torque*, dengan kontroler yang menggunakan parameter yang tepat sesuai dengan Tabel 5.1.
2. Simulasi menggerakkan lengan robot dengan metoda *computed torque*, dengan kontroler yang menggunakan parameter yang diperkecil 50% dari parameter pada Tabel 5.1.
3. Simulasi menggerakkan lengan robot dengan metoda *computed torque*, dengan perbedaan parameter tersebut di atas (seperti pada simulasi ke-2), namun menggunakan kompensator jaringan syaraf tiruan.

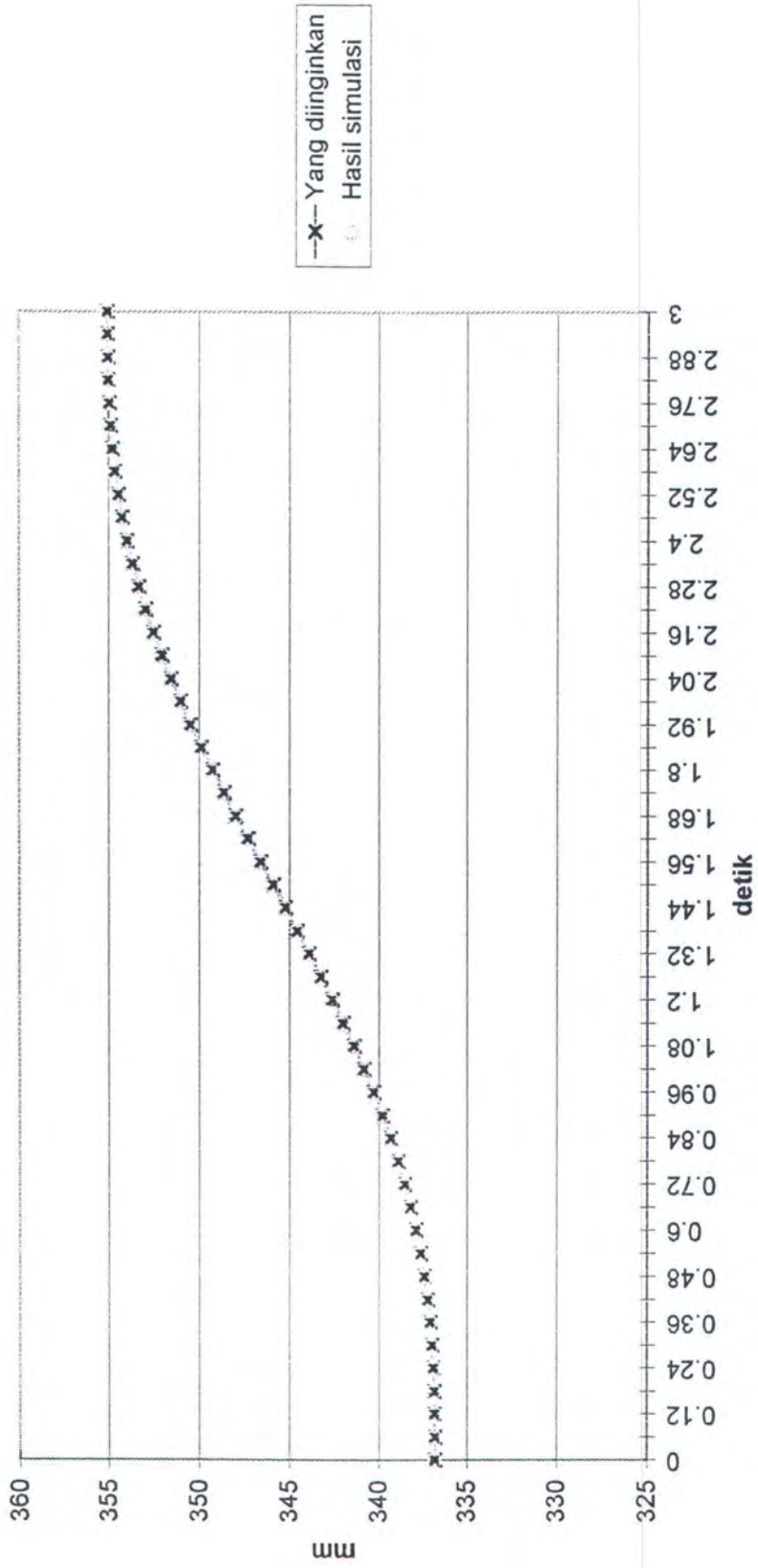
Hasil dari simulasi pertama diperlihatkan pada Tabel 5.2, dengan grafik posisi X dan Y pada Gambar 5.2. dan Gambar 5.3. Di mana posisi *end-effector* terhadap sumbu X maupun sumbu Y adalah sama dengan posisi yang diinginkan.

Hasil dari simulasi ke-2 digambarkan pada Gambar 5.4. dan Gambar 5.5, serta Tabel 5.3. Di mana posisi *end-effector* terhadap sumbu X maupun sumbu Y, terjadi kesalahan terhadap posisi yang diinginkan. Hal tersebut menjelaskan bahwa pengetahuan yang tidak tepat terhadap parameter lengan robot tersebut akan mengakibatkan kesalahan pada pergerakan lengan robot.

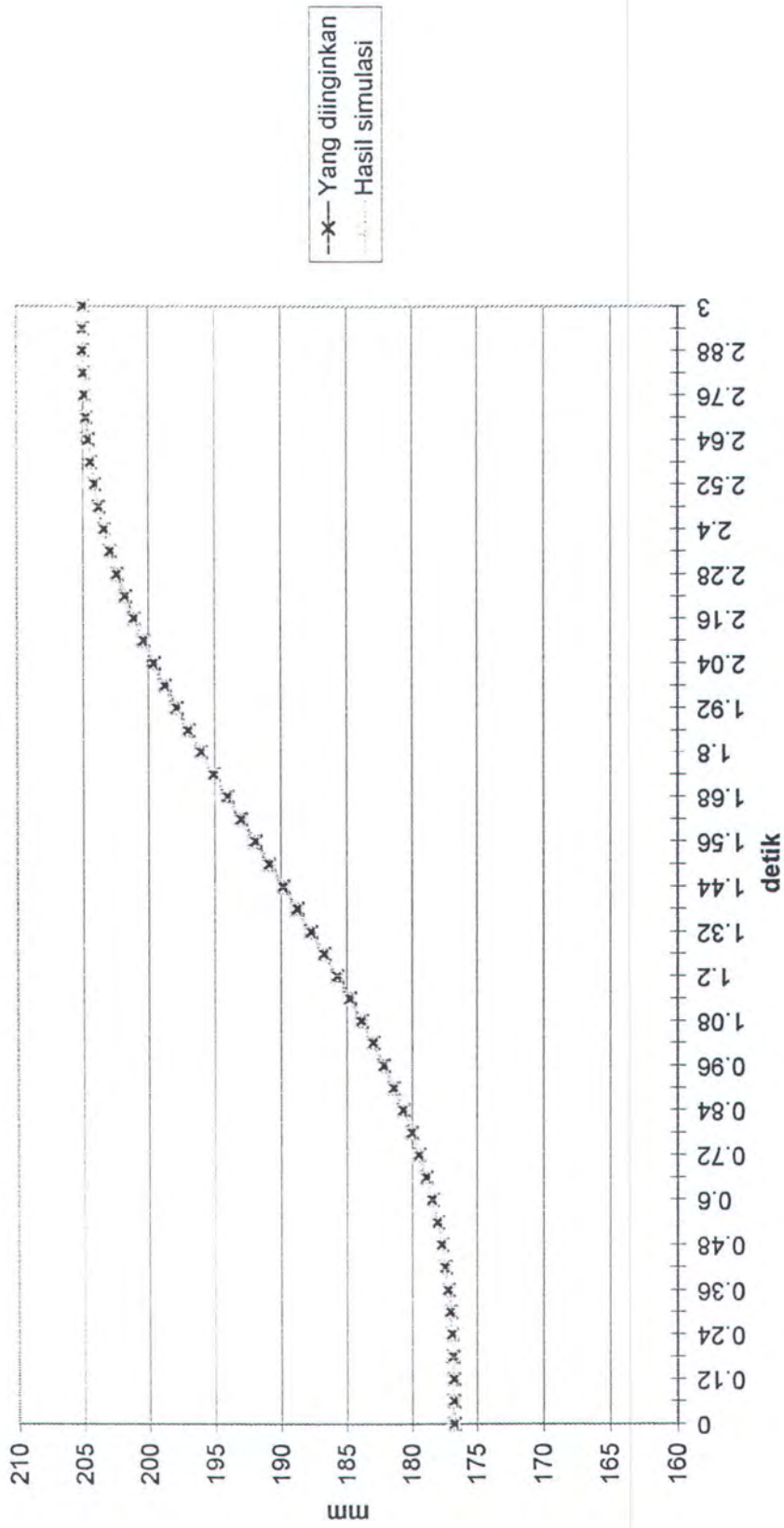
Hasil dari simulasi ke-3, diperlihatkan pada Gambar 5.6. dan Gambar 5.7, serta Tabel 5.4. Di mana posisi *end-effector* terhadap sumbu X maupun sumbu Y adalah mendekati posisi yang diinginkan. Hal tersebut membuktikan bahwa kerja dari kompensator jaringan syaraf tiruan yang diusulkan telah dapat memperbaiki kesalahan yang terjadi pada sistem. Untuk lebih jelasnya, perbandingan *error* posisi *end-effector* hasil dari simulasi ke-2 dan simulasi ke-3 diperlihatkan pada Tabel 5.5. dan Tabel 5.6.

Tabel 5.2. Hasil simulasi I

Waktu (detik)	X _d (mm)	X (mm)	Y _d (mm)	Y (mm)
0	336.7767	336.7767	176.7767	176.7767
0.3	336.9564	336.9474	177.0414	177.0316
0.6	337.8582	337.8553	178.4333	178.4306
0.9	339.7795	339.7797	181.3989	181.402
1.2	342.5996	342.602	185.7518	185.7609
1.5	345.935	345.9381	190.9	190.9136
1.8	349.2704	349.2728	196.0482	196.0635
2.1	352.0905	352.0911	200.4011	200.4151
2.4	354.0118	354.0102	203.3667	203.3768
2.7	354.9136	354.9105	204.7586	204.7637
3	355.07	355.0681	205	205.0009



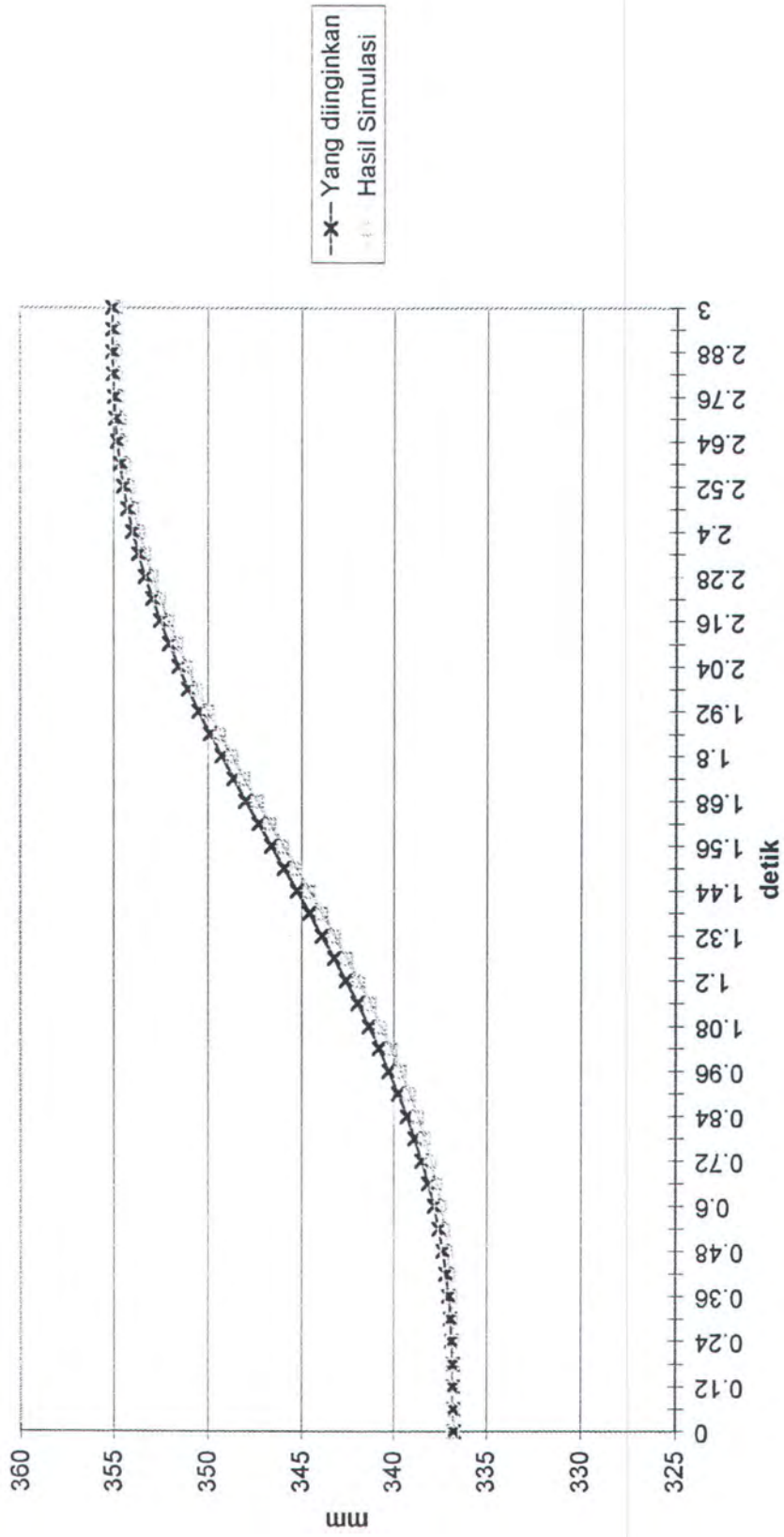
Gambar 5.2. Grafik posisi X simulasi I



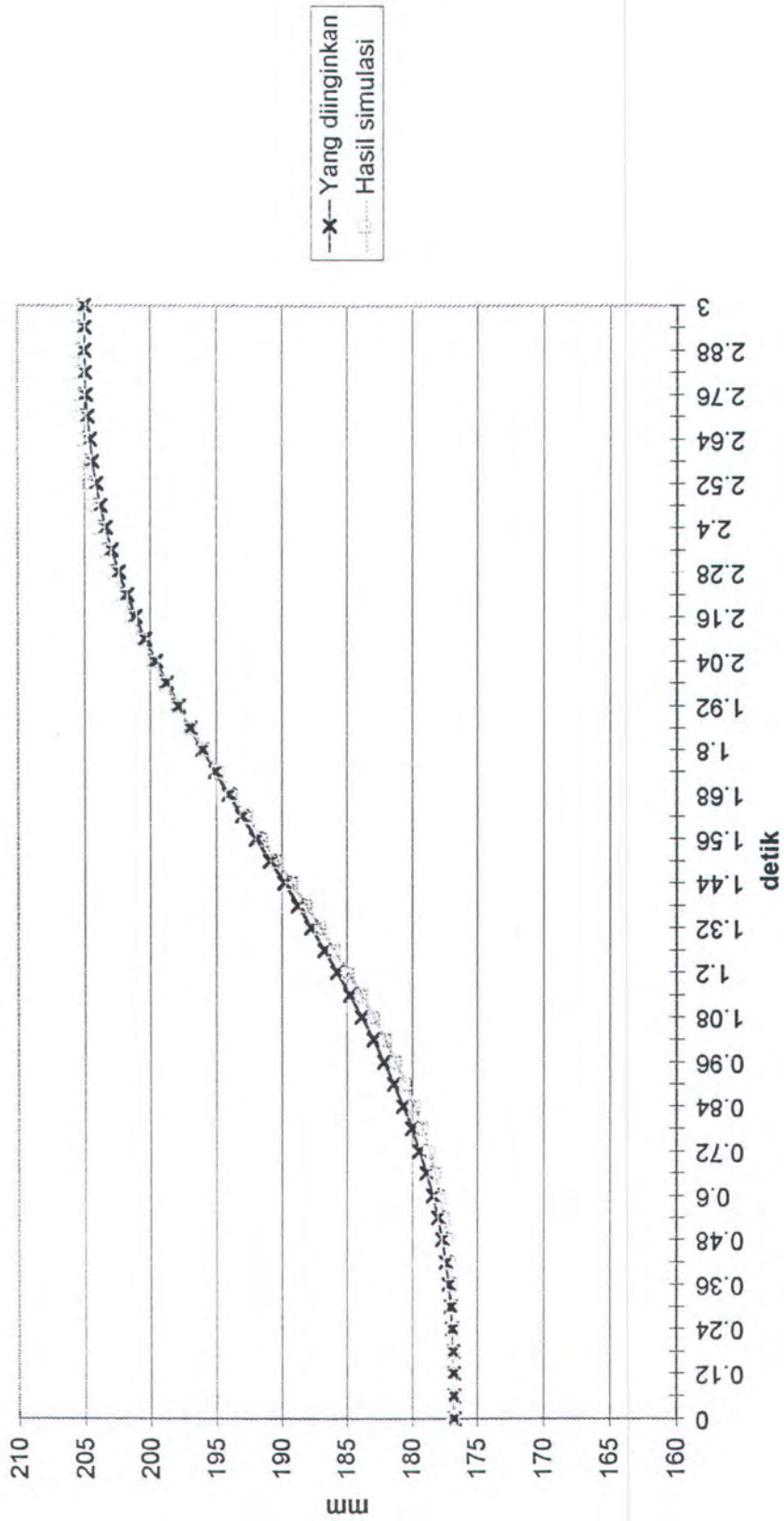
Gambar 5.3. Grafik posisi Y simulasi I

Tabel 5.3. Hasil simulasi II

Waktu (detik)	X_d (mm)	X (mm)	Y_d (mm)	Y (mm)
0	336.7767	336.7767	176.7767	176.7767
0.3	336.9564	336.8432	177.0414	176.873
0.6	337.8582	337.4526	178.4333	177.813
0.9	339.7795	339.1422	181.3989	180.4742
1.2	342.5996	341.8981	185.7518	184.8975
1.5	345.935	345.2742	190.9	190.3796
1.8	349.2704	348.6964	196.0482	195.9351
2.1	352.0905	351.6278	200.4011	200.6381
2.4	354.0118	353.669	203.3667	203.825
2.7	354.9136	354.6721	204.7586	205.2588
3	355.07	354.7865	205	205.2547



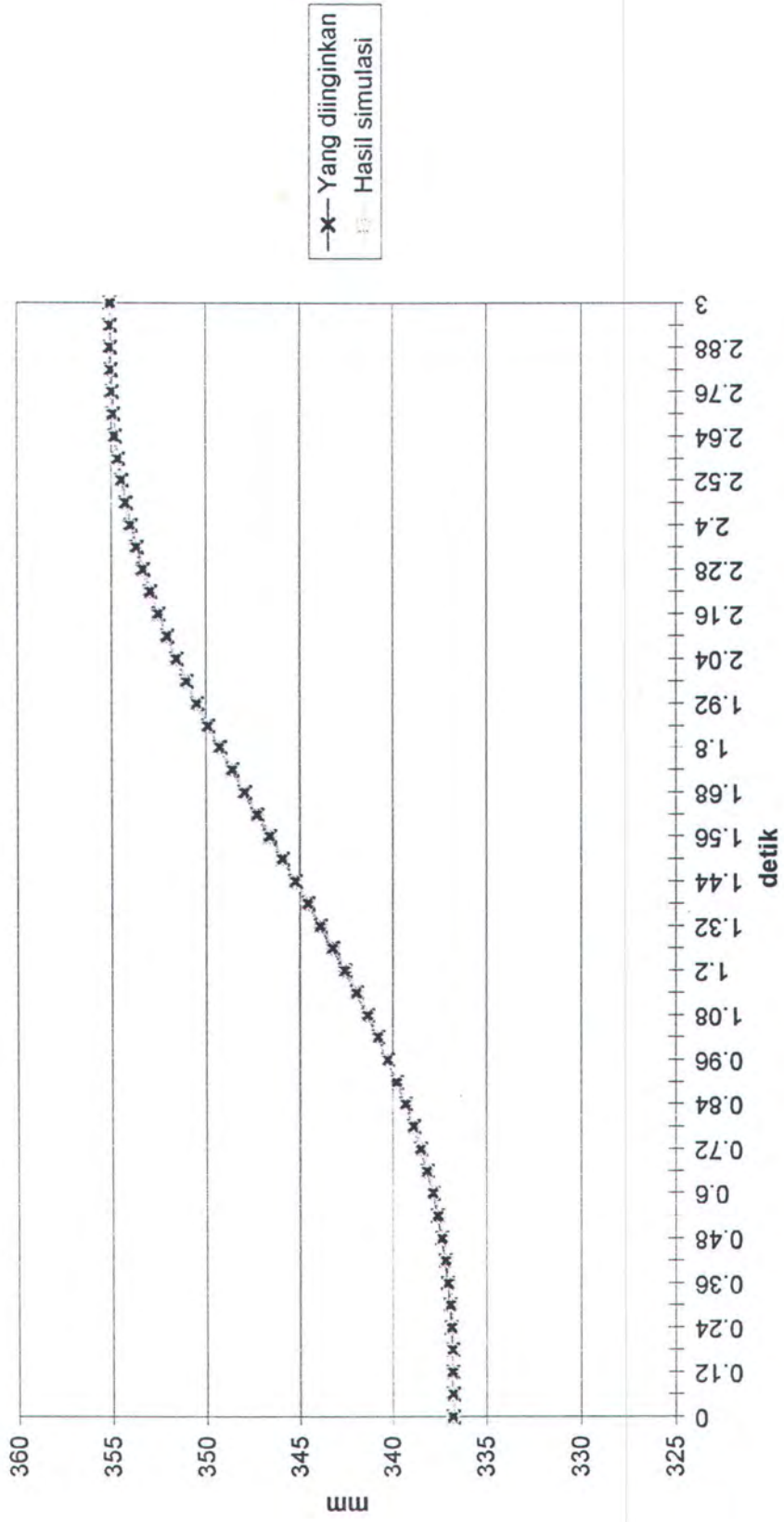
Gambar 5.4. Grafik posisi X simulasi II



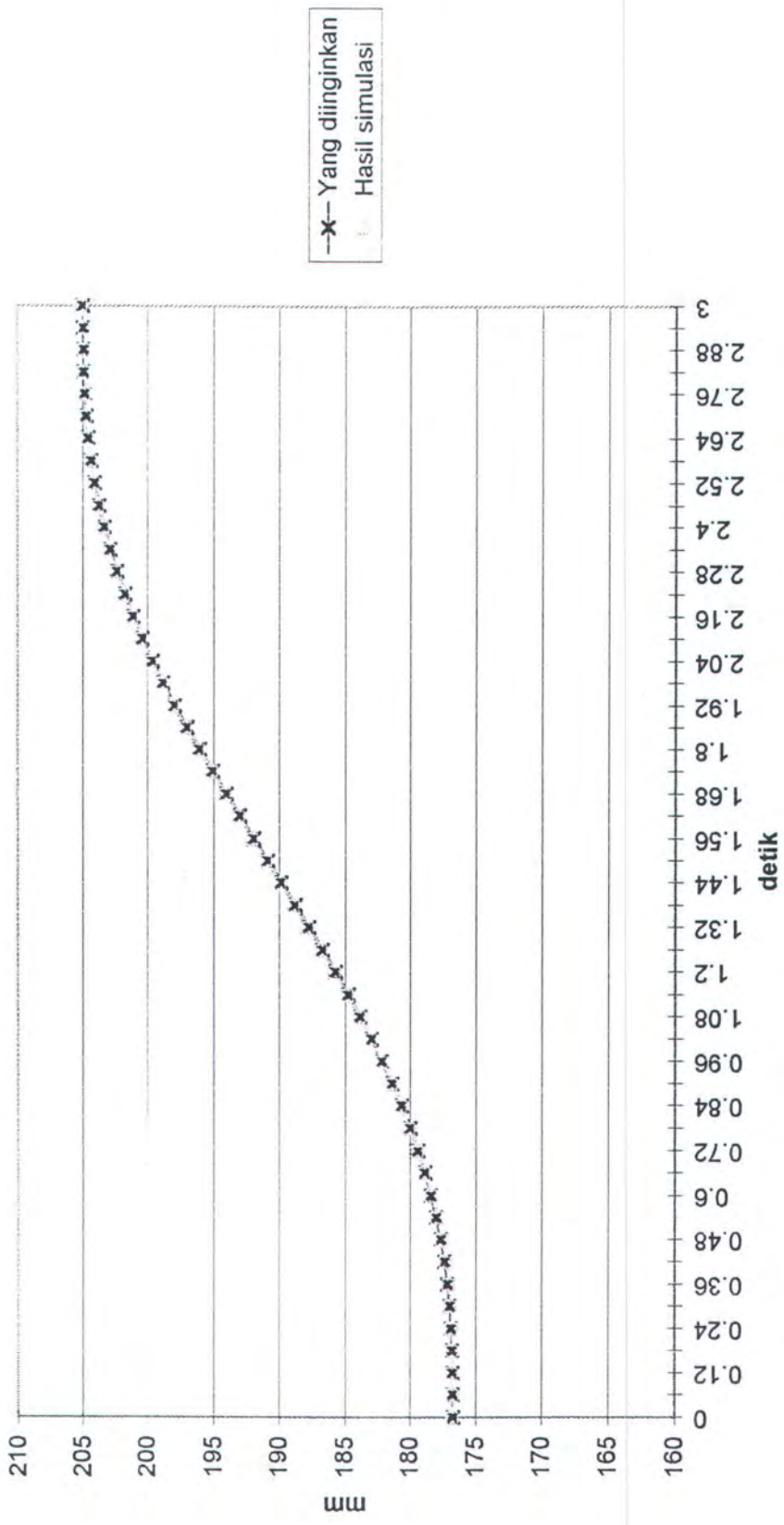
Gambar 5.5. Grafik posisi Y-simulasi II

Tabel 5.4. Hasil simulasi III

Waktu (detik)	X_d (mm)	X (mm)	Y_d (mm)	Y (mm)
0	336.7767	336.7767	176.7767	176.7767
0.3	336.9564	336.8859	177.0414	176.9368
0.6	337.8582	337.8138	178.4333	178.3821
0.9	339.7795	339.7501	181.3989	181.4011
1.2	342.5996	342.5747	185.7518	185.791
1.5	345.935	345.9195	190.9	190.9722
1.8	349.2704	349.2585	196.0482	196.1122
2.1	352.0905	352.089	200.4011	200.4658
2.4	354.0118	354.0313	203.3667	203.3942
2.7	354.9136	354.9595	204.7586	204.7404
3	355.07	354.9993	205	205.0906



Gambar 5.6. Grafik posisi X simulasi III



Gambar 5.7. Grafik posisi Y simulasi III

Tabel 5.5. Kesalahan posisi *end-effector* hasil simulasi II.

Waktu (detik)	$ X_d - X $ (mm)	$ Y_d - Y $ (mm)	$\frac{\sqrt{(X_d - X)^2 + (Y_d - Y)^2}}{l_{total}}$
0.0	0.000000	0.000000	0.000000
0.3	0.012814	0.028359	0.004058
0.6	0.164511	0.384772	0.014823
0.9	0.406151	0.855070	0.022461
1.2	0.492102	0.729828	0.022108
1.5	0.436657	0.270816	0.016822
1.8	0.329476	0.012792	0.011701
2.1	0.214091	0.056169	0.010397
2.4	0.117512	0.210039	0.011446
2.7	0.058322	0.250200	0.011109
3.0	0.080372	0.064872	0.007622

Tabel 5.6. Kesalahan posisi *end-effector* hasil simulasi III.

Waktu (detik)	$ X_d - X $ (mm)	$ Y_d - Y $ (mm)	$\frac{\sqrt{(X_d - X)^2 + (Y_d - Y)^2}}{l_{total}}$
0.0	0.000000	0.000000	0.000000
0.3	0.004970	0.010941	0.002523
0.6	0.001971	0.002621	0.001355
0.9	0.000864	4.84E-06	0.000590
1.2	0.000620	0.001537	0.000929
1.5	0.000240	0.005213	0.001477
1.8	0.000142	0.004096	0.001302
2.1	2.25E-06	0.004186	0.001294
2.4	0.000380	0.000756	0.000674
2.7	0.002107	0.000331	0.000988
3.0	0.004998	0.008208	0.002298

Bab VI

BAB VI

PENUTUP

6.1. Kesimpulan

Dari pembahasan dan simulasi yang telah dilakukan pada tugas akhir ini, dapat diambil beberapa kesimpulan sebagai berikut,

1. Pengendalian manipulator robot dengan menggunakan metoda *computed torque* hanya akan berhasil dengan baik jika parameter-parameter manipulator robot diketahui dengan tepat.
2. Ketidaktepatan pengetahuan kita terhadap parameter-parameter manipulator robot akan menimbulkan ketidakpastian (*uncertainty*) yang dapat menyebabkan kesalahan pada pergerakan dari robot tersebut.
3. Dengan menggunakan kompensator jaringan syaraf tiruan yang diusulkan, kesalahan akibat dari ketidakpastian tersebut dapat diatasi, sehingga posisi *end-effector* hasil simulasi lebih mendekati posisi yang diinginkan dengan error yang kecil sekali.
4. Metoda yang diusulkan pada tugas akhir ini secara umum efektif untuk mengatasi ketidakpastian yang mungkin terjadi. Efektifitas dari metoda yang diusulkan telah diuji untuk kasus ketidakpastian yang sifatnya statis, yaitu ketidakpastian yang terjadi akibat ketidaktepatan pengetahuan akan harga parameter manipulator robot yang sebenarnya.

6.2. Saran

Adapun beberapa saran yang perlu diberikan dalam usaha pengembangan tugas akhir ini antara lain adalah,

1. Perlu dilakukan eksperimen yang lebih banyak pada berbagai macam konfigurasi lengan robot.
2. Untuk mempercepat proses belajar dari jaringan syaraf tiruan, sebaiknya perlu dilakukan riset yang lebih jauh dalam hal struktur, algoritma, maupun pemilihan bobot awal dan konstanta perubahan bobot yang digunakan.
3. Pengujian efektifitas metoda yang diusulkan dalam tugas akhir ini akan lebih baik lagi jika diimplementasikan langsung pada manipulator lengan robot yang sesungguhnya dan untuk kasus-kasus yang mempunyai ketidakpastian yang bersifat dinamis atau berubah-ubah sebagai fungsi dari waktu.

Lampiran

RIWAYAT HIDUP



Tricahya wiratno dilahirkan di Merauke 6 Agustus 1973, sebagai anak ketiga dari empat bersaudara keluarga Bapak Walujo (Alm.). Bertempat tinggal di Jl. Kalibokor Selatan No. 154 Surabaya. Sejak tahun 1992 terdaftar sebagai mahasiswa Teknik Elektro Institut Teknologi Sepuluh Nopember Surabaya.

Riwayat Pendidikan :

- TK Teladan II Surabaya, lulus tahun 1980.
- SDN Kertajaya XII No. 218 Surabaya, lulus tahun 1986.
- SMPN 12 Surabaya, lulus tahun 1989.
- SMAN 5 Surabaya, lulus tahun 1992.

Pada bulan Maret tahun 1998 mengikuti seminar dan ujian Tugas Akhir di Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro FTI-ITS sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik Elektro.